REPORT
gem5 – NMRU

이　　름 :　　곽동우

소　　속 :　　상경대학 경제학과

학　　번 :　　2018121152

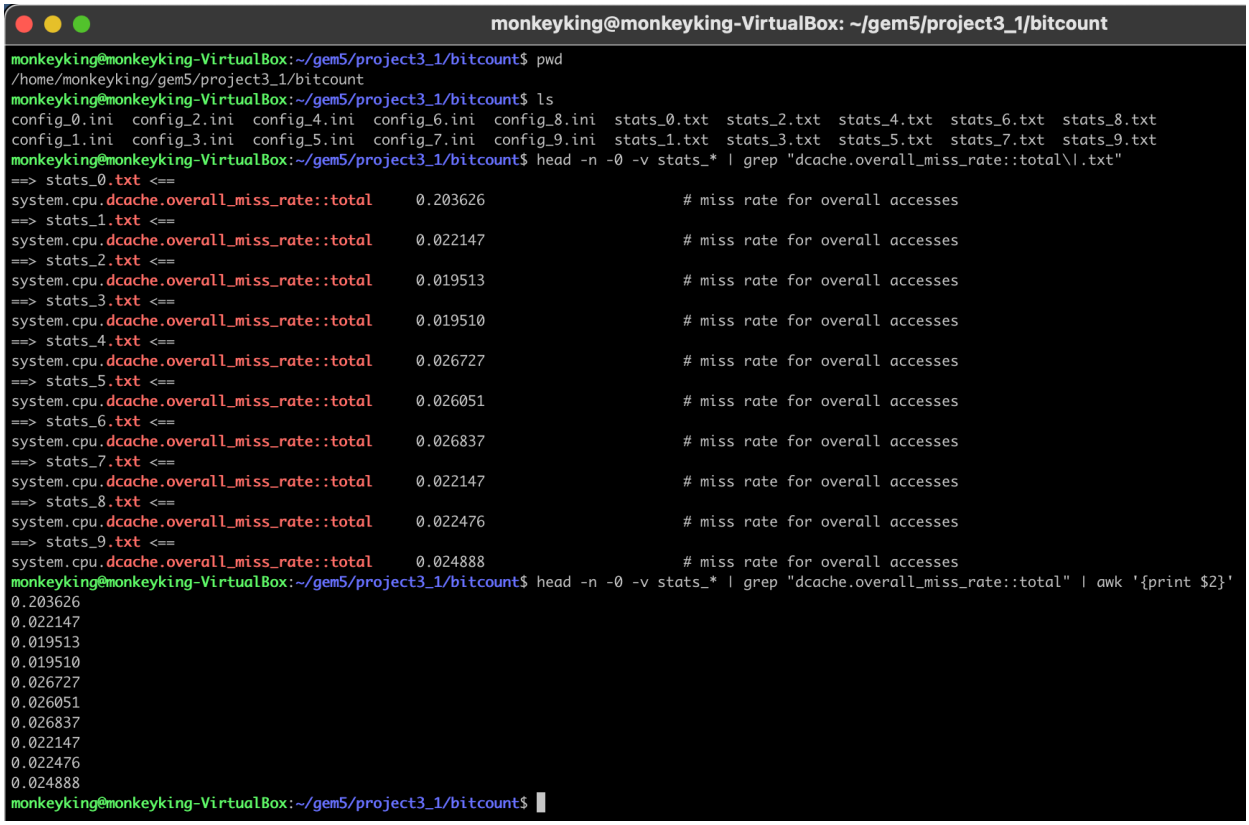
과　　목 :　　컴퓨터아키텍처

일　　시 :　　2022. 05 .26

연세대학교
YONSEI UNIVERSITY

목 차

1. 과정 스크린샷



stat_n.txt의 서식을 살펴보고 awk명령어를 통해 miss_rate(misses/hit/final_tick)만을 가져온다.

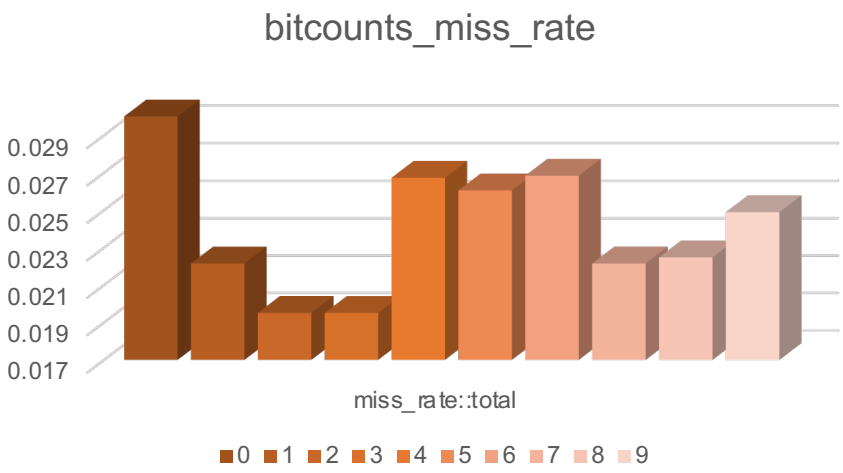
A	B	C	D	E	F	G	H
	bitcounts						
	Config #	Associativit	Replacement Policy	hits::total	misses::tot	miss_rate::tota	final_tick
	0	1	LRU (Policy 영향 X)	122648	31360	0.203626	898981500
	1	2	LRU (Default)	154935	3509	0.022147	626249500
	2	4	LRU (Default)	155268	3090	0.019513	624029500
	3	8	LRU (Default)	155240	3089	0.01951	624039500
	4	2	Random	154293	4237	0.026727	630305000
	5	4	Random	154404	4130	0.026051	628765500
	6	8	Random	154147	4251	0.026837	627316500
	7	2	MY_POLICY (NMRU)	154935	3509	0.022147	626249500
	8	4	MY_POLICY (NMRU)	154872	3561	0.022476	625691500
	9	8	MY_POLICY (NMRU)	154524	3944	0.024888	627507500

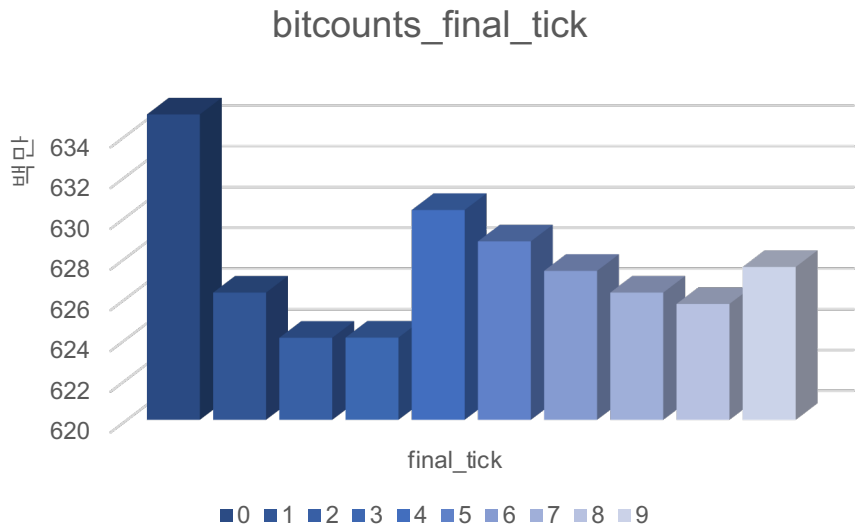
엑셀에 복사.

2. 결과분석

2. 1. bitcounts

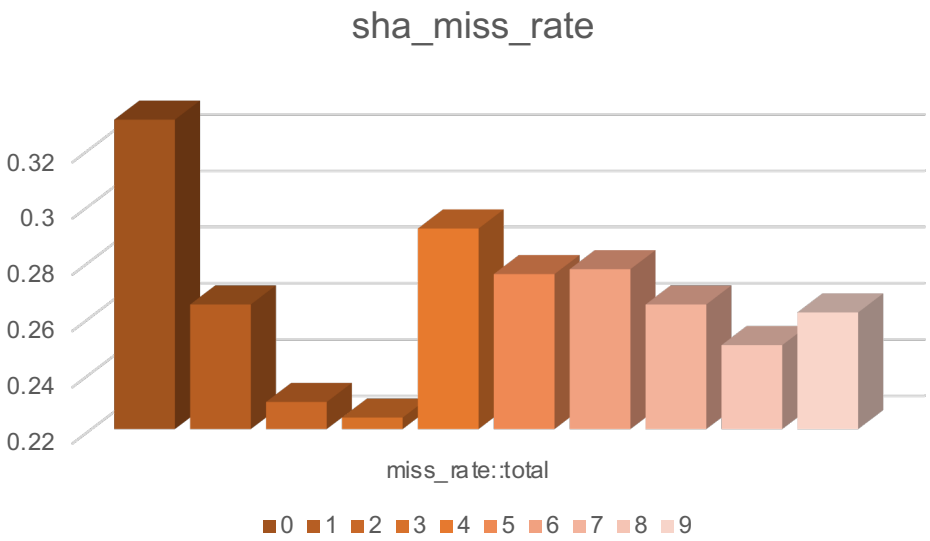
	bitcounts									
Config #	0	1	2	3	4	5	6	7	8	9
Associativity	1	2	4	8	2	4	8	2	4	8
Replacement	LRU	LRU	LRU	LRU	Random	Random	Random	MY_POLICY	MY_POLICY	MY_POLICY
hits::total	122648	154935	155268	155240	154293	154404	154147	154935	154872	154524
misses::total	31360	3509	3090	3089	4237	4130	4251	3509	3561	3944
miss_rate::total	0.203626	0.022147	0.019513	0.01951	0.026727	0.026051	0.026837	0.022147	0.022476	0.024888
final_tick	898981500	626249500	624029500	624039500	630305000	628765500	627316500	626249500	625691500	627507500

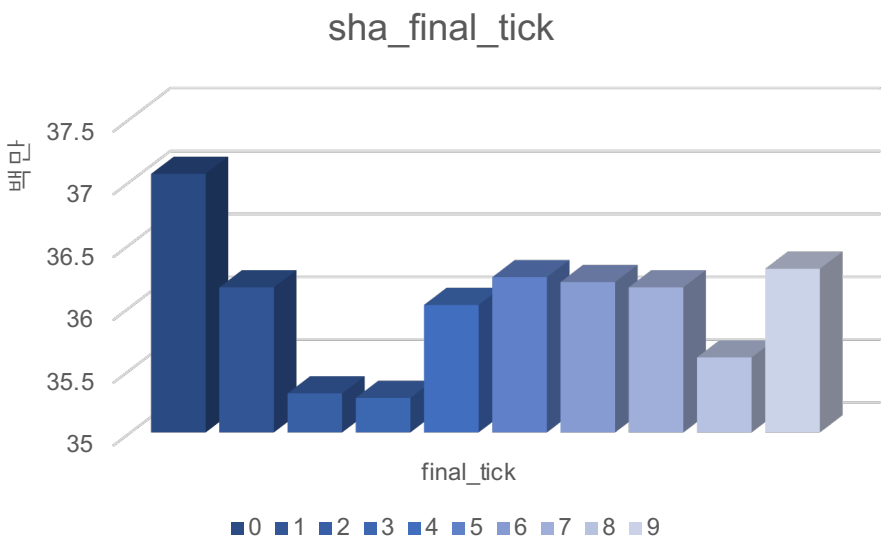




2. 2. sha

					sha					
Config #	0	1	2	3	4	5	6	7	8	9
Associativity	1	2	4	8	2	4	8	2	4	8
Replacement	LRU	LRU	LRU	LRU	Random	Random	Random	MY_POLICY	MY_POLICY	MY_POLICY
hits::total	3205	3524	3707	3729	3394	3481	3477	3524	3591	3562
misses::total	1615	1266	1105	1077	1395	1321	1331	1266	1196	1261
miss_rate::total	0.335062	0.264301	0.229634	0.224095	0.291293	0.275094	0.27683	0.264301	0.249843	0.261456
final_tick	37058000	36155000	35311000	35275000	36015000	36236000	36197000	36155000	35597000	36303000





2. 3. 성능분석

bitcounts와 sha benchmark 모두 유사한 결론의 성능을 보여주었다. 그래프에서 확인할 수 있듯이 **가장 뛰어난 성능을 보이는 configuration은 '3'**이었다. Configuration 3은 LRU Replacement Policy / Associativity = 8을 채택하고 있다. Bitcount, sha 두 벤치마크 모두에서도 가장 낮은 miss rate를 보여주고 있을 뿐만 아니라 이에 따른 성능 향상으로 tick수 또한 가장 낮을 것을 확인할 수 있다. Replacement Policy는 캐시블록의 모든 세트들이 valid상태이고 메모리의 값을 캐시에 새로이 load하기 위해 기존의 세트 중 하나를 퇴거 시켜야 할 때, 어떤 세트의 블록을 퇴거 시킬지 결정하는 정책을 일컫는다. LRU Replacement Policy는 해당 세트 블록에서 가장 오래동안 사용하지 않은 블록(Least Recently Used)을 새로 들어온 블록과 교체하는 정책이다. 프로그램의 동작 특성상 메모리의 값들은 시간적, 공간적 지역성(locality)을 띄고 있기에 현재까지 가장 사용되지 않은 캐시블록들은 앞으로도 사용이 될 확률이 극히 적을 것이다. 이러한 이유로 LRU Policy 그리고 miss_rate를 낮춰주는 Associativity가 가장 높은 Configuration3이 가장 우수한 성능(가장 낮은 Miss_rate, final_tick)을 보여준 것으로 보인다.

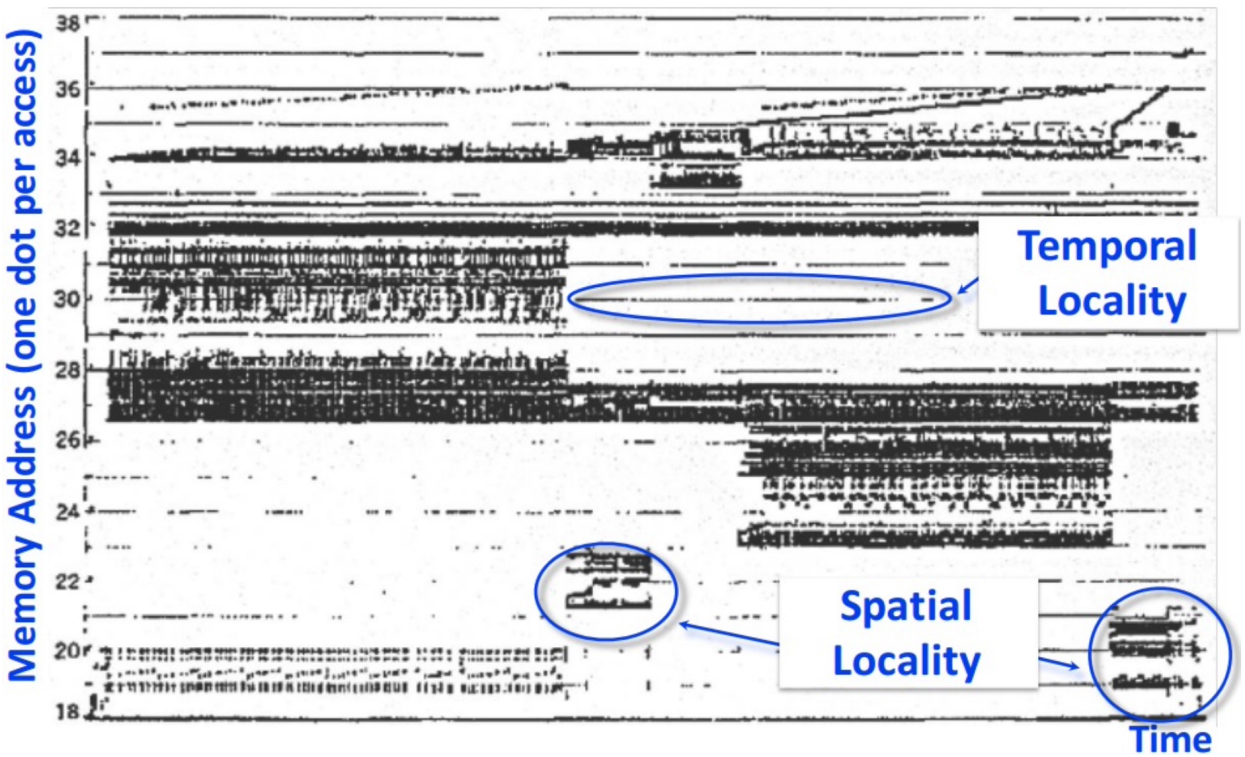


그림1. 메모리 접근의 지역성 (출처. 교재 ppt 7-1, 16pg)